

Hand-Written Digit Classification

Souvik Banerjee¹, Dr. A Rengarajan²

¹School of CS & IT, ²Professor, School of CS & IT,

^{1,2}Department of MCA, Jain University, Bangalore, Karnataka, India

ABSTRACT

Image classification is perhaps the most important part of digital image analysis. In this paper, we compare the most widely used model CNN (Convolutional Neural Network), and MLP (Multilayer Perceptron). We aim to show how both models differ and how both models approach towards the final goal, which is image classification.

KEYWORDS: Image classification, Convolutional neural network (CNN), Multilayer perceptron (MLP), Python, Pycharm, Tensorflow

How to cite this paper: Souvik Banerjee | Dr. A Rengarajan "Hand-Written Digit Classification"

Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-4, June 2021, pp.1385-1388, URL: www.ijtsrd.com/papers/ijtsrd42444.pdf



Copyright © 2021 by author (s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



INTRODUCTION

Image Classification is a fundamental task that attempts to comprehend an entire image as a whole. The goal is to classify the image by assigning it to a specific label. Typically, Image Classification refers to images in which only one object appears and is analyzed. One of the most popular applications of image classification that we encounter daily is personal photo organization. Image classification is empowering the user experience of photo organization apps. Besides offering photo storage, apps want to go a step further by giving people better search and discovery functions. Visual Search allows users to search for similar images or products using a reference image they took with their camera or downloaded from the internet.

Literature Review- CNN has been applied on the MNIST dataset in order to observe the variation of accuracies for handwritten digits. The accuracies are obtained using Tensor flow in python. Training and validation accuracy for 15 different epochs were observed exchanging the hidden layers for various combinations of convolution and hidden layers by taking the batch size 100 for all the cases.[1]

In another case, the SVC with RBF kernel (SVC-rbf) gives the highest accuracy but is extremely expensive in memory space and computation. So, the target of future classifier design is to match with the accuracy of SVC-rbf at low complexity, via extracting more discriminatory features, devising new classification/learning schemes, combining multiple classifiers, etc.[2]

Previous work performed on simple digit images (Le Cun, 1989) showed that the architecture of the network strongly

influences the network's generalization ability. Good generalization can only be obtained by designing a network architecture that contains a certain amount of a priori knowledge about the problem. The recognition is entirely performed by a multi-layer network. All of the connections in the network are adaptive, although heavily constrained, and are trained using back-propagation. The input of the network is a 16 by 16 normalized image and the output is composed of 10 units: one per class. When a pattern belonging to class i is presented, the desired output is +1 for the i -th output unit, and -1 for the other output units. [3]

METHODOLOGY:

This section presents the research methodology

The problem statement here is using two different kinds of Neural Networks to classify the same Dataset (MNIST Dataset).

MNIST is a commonly used handwritten digit dataset consisting of 60,000 images in the training set and 10,000 images in the test set. So, each digit has 6000 images in the training set. The digits are size-normalized and centered in a fixed-size (28×28) image. The task is to train a machine learning algorithm to recognize a new sample from the test set correctly.

REQUIREMENT ANALYSIS

Main tool

The main tools that drive the project are Keras and TensorFlow, as they provide the required models for the Image Classification. PyCharm, an Integrated Development Environment is used to write the Image Classifiers.

Software Requirement

Tensor Flow: Tensor Flow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

Tools Used in detail:

1. **PyCharm:** PyCharm is an integrated development environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.
2. **Python 3.8:** Python is an interpreted, high-level and general-purpose programming language. In this project, every code is in Python.
3. **Matplotlib:** Used for Plotting images.
4. **NumPy:** Used for mathematical requirements.
5. **MNIST Dataset:** The dataset used to train the models.

Architecture & Working

Architecture - Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feed forward artificial neural networks (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

An MLP (or Artificial Neural Network - ANN) with a single hidden layer can be represented graphically as follows:

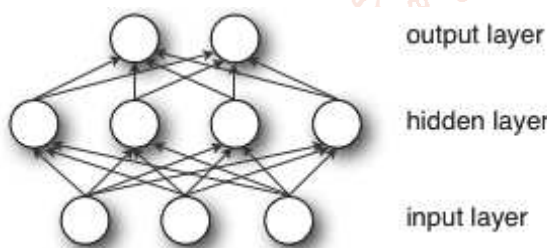


Fig: 1

Formally, a one-hidden-layer MLP is a function $f: R^D \rightarrow R^L$, where D is the size of input vector x and L is the size of the output vector $f(x)$, such that, in matrix notation:

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x))),$$

with bias vectors $b^{(1)}, b^{(2)}$; weight matrices $W^{(1)}, W^{(2)}$ and activation functions G and s .

Disadvantages of MLP include too many parameters because it is fully connected. Parameter number = width x depth x height. Each node is connected to another in a very dense web — resulting in redundancy and inefficiency.

Architecture - Convolutional Neural Network

A convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied for

analysis of visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. CNNs are regularized versions of multilayer perceptrons. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field.

Convolution is a mathematical operation that's used in single processing to filter signals, find patterns in signals etc. In a convolutional layer, all neurons apply convolution operation to the inputs, hence they are called convolutional neurons. The most important parameter in a convolutional neuron is the filter size, let's say we have a layer with filter size $5 \times 5 \times 3$. Also, assume that the input that's fed to convolutional neuron is an input image of size of 32×32 with 3 channels.

Let's pick one $5 \times 5 \times 3$ (3 for number of channels in a colored image) sized chunk from image and calculate convolution (dot product) with our filter (w). This one convolution operation will result in a single number as output. We shall also add the bias (b) to this output.

In order to calculate the dot product, it's mandatory for the 3rd dimension of the filter to be the same as the number of channels in the input. i.e. when we calculate the dot product it's a matrix multiplication of $5 \times 5 \times 3$ sized chunk with $5 \times 5 \times 3$ sized filter.

We shall slide convolutional filters over the whole input image to calculate this output across the image. In this case, we slide our window by 1 pixel at a time. In some cases, people slide the windows by more than 1 pixel. This number is called stride.

If you concatenate all these outputs in 2D, we shall have an output activation map of size 28×28 (can you think of why 28×28 from 32×32 with the filter of 5×5 and stride of 1). Typically, we use more than 1 filter in one convolution layer. If we have 6 filters in our example, we shall have an output of size $28 \times 28 \times 6$.

As you can see, after each convolution, the output reduces in size (as in this case we are going from 32×32 to 28×28). In a deep neural network with many layers, the output will become very small this way, which doesn't work very well. So, it's a standard practice to add zeros on the boundary of the input layer such that the output is the same size as input layer. So, in this example, if we add a padding of size 2 on both sides of the input layer, the size of the output layer will be $32 \times 32 \times 6$ which works great from the implementation purpose as well. Let's say you have an input of size $N \times N$, filter size is F , you are using S as stride and input is added with 0 pad of size P . Then, the output size will be: $(N - F + 2P) / S + 1$

Training and Testing Dataset:

Training data shape: (60000, 28, 28) (60000,)

Testing data shape: (10000, 28, 28) (10000,)

Total number of outputs: 10

Output classes: [0 1 2 3 4 5 6 7 8 9]

Model Summary (MLP):

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	401920
dense_1 (Dense)	(None, 512)	262656
dense_2 (Dense)	(None, 10)	5130

Total params: 669,706

Trainable params: 669,706

Non-trainable params: 0

After training on the Dataset:

Evaluation result on Test Data: Loss = 0.7112800478935242, accuracy = 0.97079998254776

Loss curve

Loss curves are a standard actuarial technique for helping insurance companies assess the amount of reserve capital they need to keep on hand to cover claims from a line of business. Claims made and reported for a given accounting period are tracked separately over time.

Below is the loss curve of the above Model:

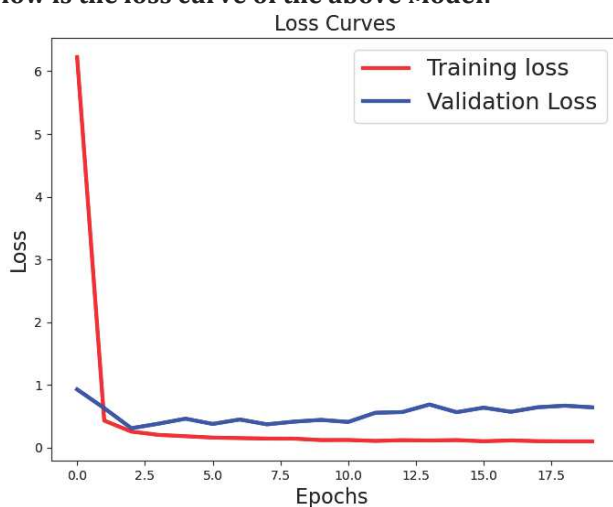


Fig. 2 Below image shows the model's prediction on an image:

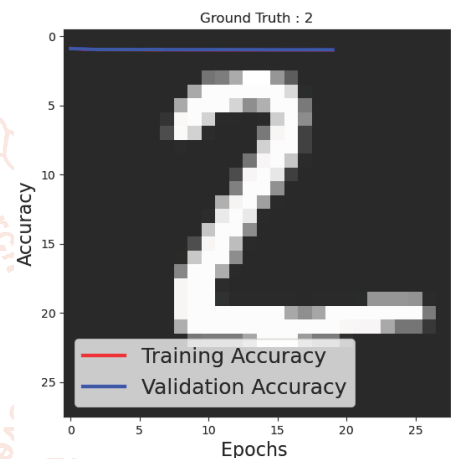


Fig. 3

Training, Testing and Validation Dataset:

Train: Found 15000 images belonging to 10 classes.

Valid: Found 1000 images belonging to 10 classes.

Test: Found 500 images belonging to 10 classes.

Model Summary (CNN):

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	896
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_1 (Conv2D)	(None, 14, 14, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 10)	31370

Total params: 50,762

Trainable params: 50,762

Non-trainable params: 0

After training on the dataset:

loss: 0.0195 - accuracy: 0.9939 - val_loss: 0.1851 - val_accuracy: 0.9600

Test batch accuracy percentage: 0.979

Confusion Matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The

confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

Below is the Confusion Matrix of the above Model:

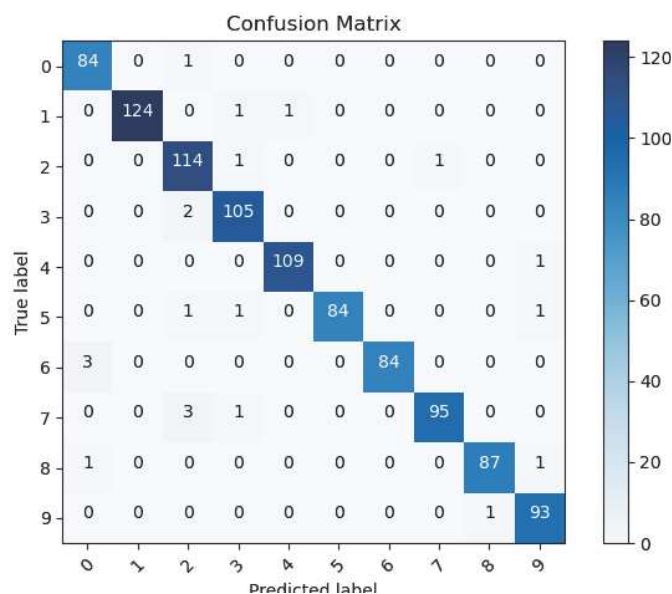


Fig. 4

Conclusion:

After Going through all the above steps, I was able to make two Image Classifiers both of which can correctly detect any hand-written digit in 10ths place.

There can be a lot of future scope in this field, a lot of features can be added to this like multi-character identification, object identification, etc.

References:

- [1] Fathma Siddique1#, Shadman Sakib2*, Md. Abu Bakr Siddique2\$, 2019. Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers, 2019 1 Department of CSE, International University of Business Agriculture and Technology, Dhaka 1230, Bangladesh, 2 Department of EEE, International University of Business Agriculture and Technology, Dhaka 1230, Bangladesh.
- [2] Cheng-Lin Liu*, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, 2003. Handwritten digit recognition: benchmarking of state-of-the-art techniques, Central Research Laboratory, Hitachi, Ltd. 1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan.
- [3] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, 1990. Handwritten Digit Recognition with a Back-Propagation Network, AT&T Bell Laboratories, Holmdel, N. J. 07733.